	Type	L #	Hits	Search Text	DBs	Time Stamp
1	IS&R	L1	297	("716/8").CCLS.	USPAT	2002/01/14 13:36
2	IS&R	L2	337	(("716/10") or ("716/11")).CCLS.	USPAT	2002/01/14 13:36
3	BRS	L3	0	floorplan with (high adj3 level adj4 language)	USPAT	2002/01/14 13:37
4	BRS	L4	55	floorplan$3 with (block$2)	USPAT	2002/01/14 13:37
5	BRS	L5	91	floorplan$3 with (block$2 or module$2 or macro$2 or unit$2 or cell$2)	USPAT	2002/01/14 13:38
6	BRS	L6	2881	716/$.ccls.	USPAT	2002/01/14 13:38
7	BRS	L8	23	1 and 5	USPAT	2002/01/14 13:48
8	BRS	L9	39	4 and 6	USPAT	2002/01/14 13:50
9	BRS	L10	57	5 and 6	USPAT	2002/01/14 13:51
10	BRS	L7	14	1 and 4	USPAT	2002/01/14 15:01
11	BRS	L11	9	8 not 7	USPAT	2002/01/14 13:48
12	BRS	L12	25	9 not 8	USPAT	2002/01/14 13:50
13	BRS	L13	25	12 not 11	USPAT	2002/01/14 13:50
14	BRS	L14	32	10 not 13	USPAT	2002/01/14 13:52
15	BRS	L15	16	floorplan$3 same (user$2 and designer$2)	USPAT; US-PG PUB; EPO; JPO; DERWENT; IBM TDB	2002/01/14 13:53

	Type	L #	Hits	Search Text	DBs	Time Stamp
1	BRS	L1	2888	716/$.ccls.	USPAT	2002/01/15 08:35
2	BRS	L2	4	generat$3 with (multi$3 or plural$3) with floorplan$2	USPAT	2002/01/15 08:43
3	BRS	L3	4	1 and 2	USPAT	2002/01/15 08:35
4	BRS	L4	5	select$3 with (multi$3 or plural$3) with floorplan$2	USPAT	2002/01/15 08:49
5	BRS	L5	3	(choose or cho$4) with (multi$3 or plural$3) with floorplan$2	USPAT	2002/01/15 08:47
6	BRS	L6	18	(select$3 or choose or chosen or chosing) with floorplan$2	USPAT	2002/01/15 08:50
7	BRS	L7	13	6 not (3 4 5)	USPAT	2002/01/15 08:50
8	BRS	L8	4	1 and 7	USPAT	2002/01/15 08:50

DOCUMENT-IDENTIFIER: EP 294188 A2
TITLE: Hierarchical floorplanning.

FPAR:
CHG DATE=19990617 STATUS=O> A system in which logic and/or memory elements are automatically placed on an integrated circuit ("floorplanning"), taking into account the constraints imposed by the logic designer, not only increases the density of the integrated circuit, and the likelihood of routing interconnections among the elements on that circuit, but it also enables the user to quickly modify the floorplan manually, and then graphically display the results of such modifications. By conforming itself to the logic designer's modular, hierarchical design, the system is capable of placing elements at each level of the specified hierarchy, based upon the number of interconnections between elements throughout that hierarchy. The system includes means for estimating the size of elements which have not yet been laid out, and for partitioning groups of elements into successively smaller "slices" of the integrated circuit (using heuristic techniques when exhaustive methods are no longer (feasible) until all elements are placed relative to one another. The system also includes means for determining the precise shapes of elements on the integrated circuit, based upon the relative placement of such elements, and

upon the additional area required for routing interconnections among such
elements. The functionality of this hierarchical floorplanning system can be
embodied in the form of software, hardware or any combination thereof, because
the system's hierarchical methodology and structure is independent of its
particular embodiment.

DOCUMENT-IDENTIFIER: US 4918614 A
TITLE: Hierarchical floorplanner

ABPL:
A system in which logic and/or memory elements are automatically placed on an integrated circuit ("floorplanning") taking into account the constraints imposed by the logic designer, not only increase the density of the integrated circuit, and the likelihood of routing interconnections among the elements on that circuit, but it also enables the user to quickly modify the floorplan manually, and then graphically display the results of such modifications. By conforming itself to the logic designer's modular, hierarchical design, the system is capable of placing elements at each level of the specified hierarchy, based upon the number of interconnections between elements throughout that hierarchy. The system includes means for estimating the size of elements which have not yet been laid out, and for partitioning groups of elements into successively smaller "slices" of the integrated circuit (using heuristic techniques when exhaustive methods are no longer feasible) until all elements are placed relative to one another. The system also includes means for determining the precise shapes of elements on the integrated circuit, based upon the relative placement of such elements, and upon the additional area required for routing interconnections among such elements. The functionality of this hierarchical floorplanning system can be embodied in the form

of
software, hardware or any combination thereof, because the system's
hierarchical methodology and structure is independent of its particular
embodiment.

BSPR:
(3) Hierarchical Interconnection of Functions By far the most significant
obstacle to achieving the optimum floorplan is the fact that optimizing the
floorplan at any given level in the hierarchy requires knowledge of functional
interconnections at other levels. Because logic and/or memory elements are
interconnected throughout the logic designer's hierarchy (not merely at the
bottom level), optimum floorplanning at any level requires a hierarchical
approach which takes these inter-level interconnections into account.

BSPR:
The system provides a graphic display of the floorplan, including the
interconnection of functions, at all or some levels of the hierarchy. Means
are also provided for the user to interact with the system in a variety of
ways, performing various parts of the floorplanning process (such as relative
placement and shape determination) manually.

BSPR:
Finally, for any particular logic design, the system supplies the user with two
percentages, one indicating the "feasibility" of laying out that design in the
specified chip area, and the other indicating, after the floorplan is complete,
the "routability" of that particular floorplan (i.e., the likelihood of

successfully routing that floorplan in the specified chip area).

DEPR:
Finally, it displays Feasibility and Routability percentages which, as discussed earlier, indicate, respectively, the "feasibility" of laying out the
user's logic design in the specified chip area, and the "routability" of the
particular floorplan of that logic design produced by the system (i.e., the
likelihood of successfully routing that floorplan in the specified chip area).

DEPR:
This routability percentage indicates, in essence, the additional manual effort
required by the user to achieve an actual layout of the logic design in the
specified chip area. The user can then, as an alternative to exerting that
effort, increase the chip area slightly and/or manually adjust the floorplan,
and then invoke the system again until a sufficiently high routability percentage is obtained.

DOCUMENT-IDENTIFIER: US 6170080 B1
TITLE: Method and system for floorplanning a circuit design at a high level of abstraction

BSPR:
An application specific integrated circuit (ASIC) is typically produced by developing a behavioral description of the desired circuit functions, determining a list of instances or <u>blocks</u> of logic needed to implement the desired functions, arranging the <u>blocks</u> of logic into a <u>floorplan</u> that meets desired constraints, and physically laying out the <u>floorplan</u> on the integrated circuit.

BSPR:
Currently, most floorplanning tools work at a relatively low, level of abstraction using structural netlists. A conventional process used to implement a circuit design is illustrated in FIG. 1a. As shown in FIG. 1a, the designer inputs a Register Transfer Level circuit design into a synthesis tool at step 100 and, using estimated wire load models, produces a netlist file (e.g., [nls]file). At step 110, the designer floorplans the netlist using a floorplanning tool. This produces a physical data electronic format file (e.g., [pdf]file) which includes a list of clusters (i.e., groups of instances representing functional <u>blocks</u> of logic), a wire load model file (e.g., [wlm]file) which includes the names of the wire load models of the physical clusters, and a floorplanning file (e.g., [flr]file) which holds the <u>floorplan</u>.

The designer then re-synthesizes the circuit design at step 120 using the wire load models, the clustering information, the netlist file produced at step 100, and timing and design rule constraint files (e.g., [tco]file and [dco]file). This re-synthesis produces a new netlist file (e.g., [nls]file') listing of which instance belongs to which cluster. Finally, the designer re-floorplans the new netlist at step 130, using the new netlist file and the floorplanning file created at step 110.

BSPR:
There is thus a need for a system and method of circuit design and implementation by which a designer can floorplan a circuit design at a high level of abstraction. This need is because the circuit designer may want to begin to floorplan the design at a high level of abstraction before finishing the netlist. For example, the designer may want to estimate the physical size of circuit design or to place certain critical entities close together to speed the signal propagation time of these entities. Alternately, the designer may want to floorplan at a high level of abstraction so that inter-block wiring capacitances and intra-block wire load models can be estimated before the circuit design is completed. This information is important for good synthesis and optimization of the circuit design. As such, a netlist of the circuit design must be produced before a system designer working at a high level of abstraction, such as the Register Transfer Level (RTL), can floorplan the circuit design.

DEPR:
In step 140 of FIG. 1b, the designer arranges a floorplan of the circuit design
at a high level of abstraction, such as the Register Transfer Level, using
estimated wire load models (e.g., [vhd]file and [v]file), timing constraints
(e.g., [tco]file), design rule constraints (e.g., [dco]file), and a
floorplanning tool implemented according to exemplary embodiments of the
present invention. This produces a parasitic file (e.g., [pst]file), which
includes the value of the estimated inter-block capacitances, a [pdf]file which
includes a list of the physical clusters, a wire load model file (e.g.,
[wlm]file), which includes the name of the wire load models of these clusters,
and a floorplanning file (e.g, [flr]file), which holds the Register Transfer
Level floorplan.

DEPR:
Before describing particular features of an exemplary floorplanning tool
according to the present invention, various terms will be discussed as a
general overview. For the purposes of this discussion, an area is the basic
block the Register Transfer Level floorplanner manipulates. An "Area" can
comprise a group of standard cells or gates. For example, a macro cell such as
a random access memory (i.e., RAM) compiled datapath is an "area".

DEPR:
According to exemplary embodiment of the present invention, the Register
Transfer Level floorplanner is a tool which can read and represent an original

logical hierarchy at a high level of abstraction, can perform **block**
placement
at the Register Transfer Level of abstraction, can permit the designer
to
manipulate a particular logical hierarchical cell without affecting the
other
hierarchical cells, can be used to break the logical hierarchy little by
little
in order to build the physical hierarchy, that is the design **floorplan,**
giving
the designer information/cues to help build the physical hierarchy, can
derive
wire load models for the physical hierarchy, can extract inter-**block**
capacitance values and communicate them to a synthesis tool in order
to drive
the optimization process, can communicate these wire load models
together with
the physical hierarchy to the synthesis tool in order to correctly drive
it,
and can write a floorplanning file which includes the size and the
location of
the different areas.

CCXR:
716/8